

WHAT WE CLAIM IS:

1. A method for providing computational services to a client using a grid-based distributed computing system, the system including a plurality of engines and at least one grid manager, the method comprising:

deploying executable code corresponding one or more service(s) such that the engines can access the executable code;

registering the service(s) with the manager;

creating instance(s) of the service(s) that may be invoked by the client; and,

using one or more of the instance(s) to invoke one or more of the registered service(s), wherein invoking a service comprises:

communicating a service request to the manager; using the manager to assign the service request to an available engine; and,

executing code corresponding to the requested service on the assigned engine.

2. A method, as defined in claim 1, wherein deploying executable code comprises storing the executable code on a shared file system accessible to the engines.

3. A method, as defined in claim 1, wherein deploying executable code comprises using a file update mechanism provided by the manager to distribute the executable code to the engines.

4. A method, as defined in claim 3, wherein using a file
5 update mechanism to distribute the executable code to the engines comprises distributing the executable code to the engines when the manager is idle.

5. A method, as defined in claim 1, wherein creating an instance of a service comprises invoking a service creation
10 method.

6. A method, as defined in claim 1, wherein communicating a service request to the manager comprises invoking a synchronous invocation method.

7. A method, as defined in claim 1, wherein communicating
15 a service request to the manager comprises invoking an asynchronous invocation method.

8. A method, as defined in claim 1, wherein communicating a service request to the manager comprises synchronously invoking a client-side proxy that corresponds to the service
20 request.

9. A method, as defined in claim 1, wherein communicating a service request to the manager comprises asynchronously invoking a client-side proxy that corresponds to the service request.

10. A method, as defined in claim 1, wherein communicating a service request to the manager comprises communicating a synchronous service request via a client-side wsdl proxy.

11. A method, as defined in claim 1, wherein communicating
5 a service request to the manager comprises communicating an asynchronous service request via a client-side wsdl proxy.

12. A method, as defined in claim 1, wherein using the manager to assign the service request to an available engine comprises applying an adaptive scheduling algorithm to select an
10 engine.

13. A method, as defined in claim 12, wherein applying an adaptive scheduling algorithm to select an engine comprises selecting among competing engines requesting work from the manager.

14. A method, as defined in claim 12, wherein applying an
15 adaptive scheduling algorithm to select an engine comprises computing a neediness score for service(s) with pending request(s).

15. A method, as defined in claim 14, wherein each
20 neediness score is computed, at least in part, from two or more of: (i) a priority weight for the service, (ii) an amount of time that engines have spent running task(s) associated with the service, and (iii) an amount of time that the request for the service has spent waiting to be assigned.

16. A method, as defined in claim 14, wherein applying an adaptive scheduling algorithm to select an engine further comprises computing affinity scores reflecting the affinities of available engines to requested service(s).

5 17. A method, as defined in claim 16, wherein each affinity score is computed, at least in part, from information representing the amount of the requested service's state that is already present on the engine-in-question.

10 18. A method, as defined in claim 14, wherein requested service(s) are first ranked according to neediness score(s), then assigned to engine(s) according to affinity score(s).

19. A method for providing a client with a flexible, service-oriented computing environment using a computing grid, the method comprising:

15 invoking a grid manager to create client-side instances of stateless services;
 invoking the grid manager to create client-side instances of stateful services;
 using one or more of the client-side instances to invoke
20 stateless services on one or more manager-assigned engines; and,
 using one or more of the client-side instances to invoke stateful services on one or more manager-assigned engines.

20. A method, as defined in claim 19, wherein invoking a grid manager to create client-side instances of stateless services comprises invoking a service creation method.

21. A method, as defined in claim 19, wherein invoking a
5 grid manager to create client-side instances of stateless services comprises creating proxy objects associated with the stateless services.

22. A method, as defined in claim 19, wherein invoking a grid manager to create client-side instances of stateful
10 services comprises invoking a service creation method.

23. A method, as defined in claim 19, wherein using one or more of the client-side instances to invoke a stateful service on one or more manager-assigned engines further comprises associating a state update method with the invoked service
15 instance.

24. A method, as defined in claim 23, wherein associating a state update method comprises associating an append state method.

25. A method, as defined in claim 23, wherein associating
20 a state update method comprises associating a set state method.

26. A method, as defined in claim 19, wherein invoking a grid manager to create client-side instances of stateful services comprises creating proxy objects associated with the stateful services.

27. A method, as defined in claim 26, further comprising implementing one or more asynchronous callback interface(s) associated with the proxy objects.

28. A method, as defined in claim 27, wherein implementing
5 an asynchronous callback interface comprises associating a response handling method with the proxy-in-question.

29. A method, as defined in claim 28, wherein implementing an asynchronous callback interface further comprises associating an error handling method with the proxy-in-question.

10 30. A method, as defined in claim 26, wherein creating proxy objects associated with the stateful services further comprises designating stateful methods.

31. A method, as defined in claim 30, wherein designating a stateful method comprises designating the method as an append
15 state method.

32. A method, as defined in claim 30, wherein designating a stateful method comprises designating the method as a set state method.

33. A method of deploying and invoking a stateful service
20 on a grid computing platform, the method comprising:
deploying service code so as to be accessible to engines in
the grid computing platform;
generating a service proxy corresponding to the deployed
service code;

configuring the proxy to include state update information
for each stateful method associated with the service
code; and,

using the proxy to synchronously and/or asynchronously

5 invoke stateful methods associated with the service
code, whereby the proxy's state update information is
used, as needed, to maintain coherent state
information between nodes of the grid computing
platform.

10 34. A method, as defined in claim 33, wherein configuring
the proxy to include state update information for each stateful
method associated with the service code comprises designating
each stateful method as an (i) append state method or a (ii) set
state method.

15